REMARKS

Claims 1-23 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(b) Rejection:

The Examiner rejected claims 1-5, 8-12, 15 and 18-23 under 35 U.S.C. § 102(b) as being anticipated by Peled (U.S. Patent 6,216,206). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Peled fails to disclose *a microprocessor, comprising both an instruction cache and a trace cache*. The Examiner cites column 3, lines 8-9, and FIG. 1, reference 105 as teaching the instruction cache of Applicants' claim and column 3, lines 26-27 and FIG. 2, reference 230 as teaching the trace cache of Applicants' claim. However, reference 105 and FIG. 2 both illustrate **the same cache memory** in the system of Peled. *See, e.g.*, column 2, lines 36-37, "FIG. 2 is a block diagram illustrating one embodiment of the cache memory of FIG. 1." This cache memory stores a plurality of trace segments in a data array and also includes trace victim cache 230, cited by the Examiner. In other words, the entire cache memory of Peled (depicted in FIG. 1 as element 105 and in FIG. 2, including its subcomponents) may be considered to be a trace cache. There is not a separate instruction cache illustrated or described in the system of Peled that would correspond to the instruction cache of Applicants' claim 1. Even if Peled could be considered to contain a separate instruction cache, as described in more detail below, it is clear that neither cache memory 105, nor the victim trace cache (TVC 230) of Peled meet the limitations of the instruction cache recited in Applicants' claims.

The Examiner in submitting that Peled teaches *a prefetch unit coupled to the instruction cache and the trace cache*, the Examiner makes the following remarks: "Note that an instruction is required to be instructions within the cache memory must be fetched from the cache to be executed; therefore, a fetch to this cache memory is considered to be a 'prefetch'. The component of the processor that completes this functionality is considered to be a 'prefetch unit.'" Applicants are unclear as to the meaning of the Examiner's remarks, and again assert that Peled does not teach a separate instruction cache memory according to the limitations of claim 1. For example, the Examiner cites Peled in column 1, lines 21-23, as teaching *wherein the prefetch unit is configured to fetch instruction code from a system memory for storage within the instruction cache*. However, this passage describes the operation of a type of cache memory found in the prior art that is explicitly described as distinct from the type of cache memory found in the system of Peled. The cache memory described in this passage is one that is not organized by instruction trace segments, but by instruction addresses, "In a cache organized by instruction address, a full line of adjacent instructions is typically fetched from the main memory and loaded into the cache. If the cache becomes fill, an existing line in the cache memory is replaced to accommodate a new line of instructions required by the microprocessor. The replacement of a particular line does not impact any other lines in the cache." **The system of Peled does not include this type of cache, but instead includes a cache memory organized by trace segments, and storing trace segments and trace segment members in two components.** See, e.g., Peled's Abstract, "A cache memory includes a data array and a trace victim cache. The data array is adapted to store a plurality of trace segments. Each trace segment includes at least one trace segment member. The trace victim cache is adapted to store plurality of entries. Each entry includes a replaced trace segment member selected for replacement from one of the plurality of trace segments…" (emphasis added).

Neither cache memory 105 nor the victim trace cache (VTC 230) of Peled can be interpreted to be the instruction cache of Applicants' claims, as they do not meet the limitations recited for the instruction cache. For example, cache memory 105 is not loaded by a prefetch unit fetching instruction code from a system memory, but is loaded by a complex trace build process. Similarly, TVC 230 is not loaded by a prefetch unit fetching instruction code from a system memory, but instead receives trace segment members removed from the data array 200 of Peled's cache memory 105. In addition, nothing in Peled describes fetching or storing lines of instructions into TVC 230, as recited in claim 1. The Examiner's cited passage is not directed to cache memory 105 or to TVC 230, but to a prior art instruction cache memory that is organized by instruction address and loaded one full line at a time from the main memory. **Applicants assert that Peled's system does not include such an instruction cache at all, much less both an instruction cache and a trace cache, according to the limitations recited in claim 1.**

Further regarding claim 1, Peled fails to disclose *wherein the prefetch unit is further configured to fetch a line of instructions into the instruction cache in response to a trace being evicted from the trace cache.* The Examiner cites column 7, lines 50-52 as teaching this limitation. This passage states, "To facilitate later retrieval, the replaced trace segment member is stored in the TVC 230." In other words, this passage describes that a replaced trace segment member (which may be considered an evicted trace segment member) is replaced by a new entry in the data array of Peled's cache memory and stored in another component of Peled's cache memory (the trace victim cache). **The Examiner's interpretation of this passage in teaching this limitation of claim 1 is not consistent with his own earlier interpretation of the limitations of claim 1.** For example, in earlier remarks, the Examiner submitted that Patel's cache memory 105 corresponds to Applicants' instruction cache and that Patel's cache memory subcomponent (trace victim cache 230) corresponds to Applicants' trace cache. However, in his later remarks, the Examiner interprets storing a replaced trace segment member in trace victim cache 230 as fetching instructions into the instruction cache. **Applicants again assert that neither cache memory 105 nor trace victim cache 230 could be considered to be the instruction cache of Applicants' claims. In addition, Peled teaches absolutely nothing about *fetching a line of instructions into an instruction cache in response to a trace being evicted from the trace cache*, as required by claim 1, in the cited passage or elsewhere.** As discussed above, no such separate instruction cache exists in Peled and no such fetching a line into such a cache (or any cache) is performed in the system of Peled in response to eviction of a trace (or any portion thereof) being evicted.

Anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical invention must** be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Peled fails to disclose a microprocessor having both an instruction cache and a trace cache, and that operates in the manner recited in Applicants' claim 1. Therefore, Peled cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested.

Independent claims 8, 15, and 23 include limitations similar to those of claim 1, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 7, contrary to the Examiner's assertion, Peled fails to disclose *wherein the prefetch unit is configured to inhibit the fetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the evicted trace is predicted unlikely to re-execute*. The Examiner cites column 7, lines 42-46 as teaching these limitations. This passage states, "At some point in time the cache memory 105 will need to replace an existing data line to store a new trace segment, thereby replacing a trace segment member of an existing trace segment Candidates for replacement may be chosen by way prediction, least recently used (LRU), and the like." In other words, this passage describes that trace segment members may be selected for removal (or eviction) from cache memory 105 by way prediction. **It teaches absolutely nothing about inhibiting the fetch of a line of instructions into an instruction cache (or into any cache) in response to an evicted trace (or portion thereof) being predicted unlikely to re-execute, as required by claim 7.** In addition, as discussed above regarding claim 1, Peled does not teach the instruction cache of Applicants' claims, or anything about the operation thereof. Therefore, Peled cannot be said to anticipate claim 7.

For at least the reasons above, the rejection of claim 7 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 14 and 18 include limitations similar to those of claim 7, and so the arguments presented above apply with equal force to these claims, as well.

Contrary to the Examiner's assertion, Peled fails to disclose *wherein the prefetch unit is configured to fetch a line into the instruction cache comprising instructions that correspond to operations that precede a branch in the evicted trace* (as recited in claim 2) and *wherein the prefetch unit is configured to fetch a line into the instruction cache comprising instructions that correspond to operations that follow a branch in the evicted trace* (as recited in claim 3). The Examiner cites column 1, lines 12-42, as teaching these limitations. This passage describes two different cache memory organizations according to prior art methods, one organized by instruction addresses and one organized by trace segments. **These is nothing in the description of these cache memories about <u>fetching a line of instructions comprising instructions</u> having anything to do with <u>an evicted trace</u> or operations <u>preceding or following a branch included in an evicted trace</u>.** In addition, as discussed above regarding claim 1, Peled does not teach the instruction cache of Applicants' claims, or anything about the operation thereof. Therefore, Peled cannot be said to anticipate claims 2-3.

For at least the reasons above, the rejection of claims 2-3 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 9-10, and 19-20 include limitations similar to those of claims 2-3, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 4, contrary to the Examiner's assertion, Peled fails to disclose *wherein the prefetch unit is configured to prefetch a plurality of lines of instructions into the instruction cache in response to the trace being evicted from the trace cache.* The

Examiner cites column 2, lines 15-17, and column 1, lines 21-23, as teaching these limitations. These passages state, respectively, "Each entry includes a replaced trace segment member selected for replacement from one of the plurality of trace segments" and "In a cache organized by instruction address, a full line of adjacent instructions is typically fetched from the main memory and loaded into the cache." The first passage describes the entries in victim trace cache 230, which, as discussed above in remarks regarding claim 1, <u>does not meet the limitations recited for the instruction cache of Applicants' claims</u>. For example, victim trace cache 230 is not loaded by a prefetch unit fetching a line of instructions from a system memory, as recited in claim 1. The second passage describes the operation of an instruction cache <u>in a system other than Peled's</u>. No such instruction cache is included in the system disclosed by Peled. The description in this passage illustrates a type of cache memory organized by instruction address and loaded by fetching a line of instructions from main memory. This passage does not describe the operation of the victim trace cache 230 of Peled, which the Examiner erroneously equates with the instruction cache of Applicants' claims. **In addition, this passage teaches nothing about prefetching <u>a plurality of lines of instructions</u> into an instruction cache (or any cache), as required by claim 1, much less <u>in response to a trace being evicted from a trace cache</u>.** Therefore, Peled cannot be said to anticipate claim 4.

For at least the reasons above, the rejection of claim 4 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 11 and 21 include limitations similar to those of claim 4, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 5, contrary to the Examiner's assertion, Peled fails to disclose *wherein the prefetch unit is configured to fetch a number of lines that is proportional to the number of branch operations comprised in the evicted trace*. The Examiner again cites column 1, lines 12-42 as teaching these limitations. As discussed above, this passage describes two different cache memory organizations, including one that fetches instructions into a cache memory from a main memory. This passage does not describe the operation of the victim trace cache 230 of Peled, which the Examiner erroneously equates with the instruction cache of Applicants' claims. **In addition, nothing in this passage, or elsewhere in Peled, describes anything about the <u>number of lines fetched</u> into an instruction cache (or any cache) having anything to do with <u>an evicted trace</u>, much less with <u>a number of branches in an evicted trace</u>, as required by claim 5.** Therefore, Peled cannot be said to anticipate claim 5.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 12 and 22 include limitations similar to those of claim 5, and so the arguments presented above apply with equal force to these claims, as well.

<u>Section 103(a) Rejection</u>:

The Examiner rejected claims 6, 7, 13, 14, 16 and 17 under 35 U.S.C. § 103(a) as being unpatentable over Peled in view of prior art. Applicants traverse this rejection for at least the following reasons.

Regarding claim 6, contrary to the Examiner's assertion, Peled fails to teach or suggest *wherein the prefetch unit is configured to inhibit the fetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the line of instructions is already stored in the instruction cache.* The Examiner admits that Peled fails to disclose a cache that checks for duplicate information within the cache to inhibit storing the duplicate information. The Examiner takes Official Notice that it would have been obvious at the time of the invention for one of ordinary skill in the art to take the invention of Peled and utilize a cache system that, when presented with information to store, checks if that information already exists and inhibits duplicate storage. The Examiner submits that within the invention of Peled, the combination would check when there is an eviction from the trace cache. The Examiner submits that this technique is extremely common practice within cache systems and that storing the same information within different portions of a cache is a waste of resources that can create a detriment to the processing system with regards to space, cost, power and speed.

Pursuant to M.P.E.P. § 2144.03, Applicants traverse the Examiner's taking of Official Notice. Applicants assert that while some cache systems may include a feature that prohibits storage of duplicate information, there is nothing in any art of record describing that such a feature is inherent or well known in the particular cache memory system of Peled, or that the lack of such a feature would be detrimental to the cache memory of Peled in any of the ways listed by the Examiner. Applicants assert that some cache memory systems have perfectly valid, performance-related reasons for allowing, or even encouraging, the storage of some duplicate information in an instruction cache and/or in a trace cache. For example, in some systems, duplication of at least a portion of some traces in a trace cache may allow for better performance by allowing both the predicted-taken and predicted-not-taken targets of each branch to be included in a trace cache. Other systems may allow or encourage duplicate information for other reasons. Applicants assert that the above-referenced feature of claim 6 is not well known is the type of cache system described by Peled nor in the system as recited in Applicants' claim 1. Accordingly, Applicants traverse the Examiner's taking of Official Notice and his reasoning to combine such a feature with the system of Peled. There in no evidence of record that supports the Examiner's assertions. Pursuant to M.P.E.P. § 2144.03 Applicant asserts that "the examiner must provide documentary evidence in the next Office action if the rejection is to be maintained." *See also* 37 CFR 1.104(c)(2), (d)(2) and *In re Zurko*, 258 F.3d 1379, 1386 (Fed. Cir. 2001).

Also regarding claim 6, this claim does not merely recite that a cache memory is presented with duplicate information that it determines not to store, but instead recites that *the prefetch unit is configured to inhibit the fetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the line of instructions is already stored in the instruction cache.* In other words, in Applicants' claimed invention, duplicate information is not presented to the instruction cache (i.e., after being fetched) and checked to see that it duplicates information already present. Instead, the line of instructions is not fetched at all, in the case recited in claim 6.

Further regarding claim 6, Peled in view of prior art, does not teach or suggest anything about *inhibiting the fetch of a line of instructions into the instruction cache* ***in response to the eviction of certain traces*** *from the trace cache*, as recited in the claim, nor has the Examiner included anything in his remarks explaining how he believes the references teach this limitation. **The Examiner's statement, "within the invention of Peled, the combination would check when there is an eviction from the trace cache" is completely unsupported by the cited art, as Peled does not describe fetching of instructions into an instruction cache in response to an eviction from the trace cache at all.** Therefore, there would be no reason to inhibit such fetching. In addition, as discussed above regarding claim 1, Peled does not teach the instruction cache of Applicants' claims, or anything about the operation thereof. Therefore, it is not clear how the Examiner believes a feature for not duplicating information in an instruction cache could be included in the system of Peled, since Peled's system does not include this instruction cache. Such an addition would clearly not be sufficient to teach all of the limitations of claim 6.

Applicants remind the Examiner that to establish a *prima facie* obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. Obviousness cannot be established by combining or modifying the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion or incentive to do so. *In re Bond,* 910 F. 2d 81, 834, 15 USPQ2d 1566, 1568 (Fed. Cir. 1990). As discussed in detail above, the cited reference, taken alone or in combination with what is known in the prior art, does not teach or suggest all limitations of the currently pending claims, nor has the Examiner provided sufficient motivation to combine what is known in the prior art with the cited reference. **Therefore, the Examiner has failed to state a *prima facie* rejection of claim 6.**

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 13 includes limitations similar to those of claim 6, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 7, contrary to the Examiner's assertion, Peled fails to teach or suggest *wherein the prefetch unit is configured to inhibit the fetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the evicted trace is predicted unlikely to re-execute*. The Examiner states, "Note that, as explained in the reference, a victim cache takes items removed from the main cache. The algorithm may later place that item back in the main cache and, later yet, remove it again. In this circumstance the cache would check the victim cache and inhibit the fetching of this item, because it is already there." Applicants again assert that the victim trace cache of Paled clearly does not meet the limitations recited for the instruction cache of Applicants' claims. References to this victim trace cache teach nothing about the operation of the instruction cache of Applicants' claims. In addition, the Examiner's statement "In this circumstance the cache would check the victim cache and inhibit the fetching of this item, because it is already there" is completely unsupported by the cited art. In fact, Peled describes that the victim trace cache is not checked for a match <u>unless a trace segment is not found in the data array of the cache memory</u>, "If the head lookup process results in a cache miss, the execution mode state machine queries the TVC 230 with the NLIP 440 to determine if a corresponding TVC entry 900 is present by matching the LIP 910 of the TVC entry 900."

Furthermore, claim 7 has nothing to do with the detection of a duplicate entry in the trace cache or the instruction cache. Instead, it involves the inhibiting of a fetch of a line of instructions into the instruction cache <u>in response to the eviction of certain traces from the trace cache if the evicted trace is predicted unlikely to re-execute</u>. There is nothing in Peled or any other art of record that teaches or suggests this limitation.

Applicants note that the Examiner has rejected claim 7 under both 35 U.S.C. § 102(b) and 35 U.S.C. § 103(a), citing different portions of Peled in remarks regarding the two rejections. However, for at least the reasons presented above and in remarks regarding the rejection under 35 U.S.C. § 102(b), the rejections of claim 7 are unsupported by the cited art and removal thereof is respectfully requested.

Claim 14 includes limitations similar to those of claim 7, and was also rejected under both 35 U.S.C. § 102(b) and 35 U.S.C. § 103(a), and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 16, contrary to the Examiner's assertion, Peled fails to teach or suggest *checking the instruction cache for lines of instructions comprising the instructions corresponding to the evicted trace*. The Examiner states, "Noting *[sic]* that the instruction cache accepts victim traces upon removal (col. 7 lines 50-52) and that the cache checks for duplicate entries (as combined), the limitation above appropriately follows." First, as discussed at length herein, **TVC 230 clearly does not meet the limitations recited for the instruction cache.** The Examiner's cited passage states, "To facilitate later retrieval, the replaced trace segment member is stored in the TVC 230." In other words, when a trace segment member is removed from the data array of the cache memory in Peled, it is always stored in TVC 230. There is nothing in Peled that teaches or suggests checking TVC 230 for a line of instruction comprising the instructions corresponding to an evicted trace. In fact, TVC 230 is not described as storing or receiving lines of instructions at all. Instead, it is described as receiving individual trace segment members replaced in the data array of Peled's cache memory. As discussed above in remarks regarding claim 6, there is nothing inherent about any particular cache memory checking for duplicate entries, nor has the Examiner provided any evidence that TVC 230 would benefit from such a feature. In addition, it is not clear whether TVC 230 could ever already contain a replaced trace segment member in the system of Peled, so that it would need to be checked.

For at least the reasons above, the rejection of claim 16 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claim 17, contrary to the Examiner's assertion, Peled fails to teach or suggest *inhibiting the fetching of the line of instructions into the instruction cache if the line of instructions is stored in the instruction cache*. The Examiner states only, "The functionality of claim 17 follows the combination as described above." Applicants assume the Examiner means to reference his remarks regarding claims 6 and 16. However, as discussed above, the rejection of these claims is unsupported by the cited art and the Examiner's remarks. Therefore, the arguments presented above regarding claims 6 and 16 apply with equal force to this claim, as well.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-91600/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: February 13, 2008